CS2040C Data Structure and Algorithm Assignment #1

A zip file of the solution files for MS Visual Studio (or .zip of XCode file for Mac users) is provided that contains

- The Linked List class mentioned in lecture
  - `LinkedList.h`
  - `LinkedList.cpp`
- A main file to use the Linked List:
  - `main.cpp`

You should NOT modify `LinkedList.h`. You only need to submit the code inside `LinkedList.cpp`. Feel free to change the test cases in main.cpp but you do not need to submit it.

You have to implement the bodies of the member functions of the List:

- `int List::head()`
  - return the head item of the list
- `bool List::empty()`
  - check if the list is empty or not
- `size_t List::size()`
  - return the number of items in the list
- `void List::push_head(int element)`
  - insert int input `element` into the list at the head like what the lecture mentioned
- `int List::pop_head`
  - remove and return the head item like what the lecture mentioned
- `bool List::contains(int element)`
  - check if `element` is in the list
- `std::string List::to_string()`
  - return a string of the list in the following format
  - The string should start and end with "{" and "}" with commas ", " as separators. There is a space after each comma. There is no space and comma before the last "}". The list contains nothing, the string should be just "{}". You string does not need to include the double quotes.

This is the sample output of your given code if you implemented it correctly:

```
push_head_test1()
{3, 9, 7, 9, 8, 5, 3, 5, 6, 2, 9, 5, 1, 4, 1, 3}
pop_head_test1()
{6, 2, 9, 5, 1, 4, 1, 3}
```

Submission

Please copy your LinkList.cpp into the box in Couresmology. You can but you don't have to include the line "`#include "LinkList.h""`.

Error Handling

Other than the above implementation, you should consider special erratic situations. You should throw an "`out_of_range`" exemption if such situation happens.

Extra Challenges

We will not grade this but you can think of how to implement

- Inserting an item at the tail of the list
- Returning the item at the tail of the list.